PCT/DE2004/001326

BUNDESREPUBLIK DEUTSCHLAND

Prioritätsbescheinigung über die Einreichung einer Patentanmeldung

Aktenzeichen: 103 28 241.6

Anmeldetag: 24. Juni 2003

Anmelder/Inhaber: Robert Bosch GmbH, 70469 Stuttgart/DE

Bezeichnung: Spezifikation des Ablaufs des Software-Updates für elektronische Steuergeräte durch Flash-Programmierung über serielle Schnittstellen

IPC: G 11 C, B 60 R

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 15. Juli 2004
Deutsches Patent- und Markenamt
Der Präsident
Im Auftrag

Stanschus

A 9161
06/00
EDV-L

R. 306257

20.06.03 Sy

ROBERT BOSCH GMBH, 70442 Stuttgart

Spezifikation des Ablaufs des Software-Updates für elektronische Steuergeräte durch Flash-Programmierung über serielle Schnittstellen

Stand der Technik

Der Einsatz von Flash als Speichertechnologie für Programm- und Datenstand nimmt in Seriensteuergeräten zu. Dies ermöglicht Software-Updates für Steuergeräte im Feld durch Neuprogrammierung des Flash-Speichers über serielle Schnittstellen, zum Beispiel mit einem Flash-Programmierwerkzeug über die zentrale Off-Board-Diagnoseschnittstelle des Fahrzeugs. Damit ist ein Software-Update ohne Ausbau des Steuergerätes aus dem Fahrzeug möglich, was zu erheblichen Kosteneinsparungen gegenüber einem Steuergeräteaustausch führt.

Bei dieser Art der Flash-Programmierung sind insbesondere im Service der Fahrzeuge und im Bereich sicherheitsrelevanter elektronischer Steuergeräte hohe Sicherheits- und Zuverlässigkeitsanforderungen zu erfüllen.

Beschreibung der Erfindung mit Vorteilen

Nach einigen technischen Randbedingungen der Flash-Programmierung wird im folgenden ein möglicher Ablauf für die Flash-Programmierung von Steuergeräten über die serielle Off-Board-Diagnoseschnittstelle prinzipiell dargestellt.
Für die Spezifikation und den Test des Zusammenspiels zwischen Flash-Programmierwerkzeug und Steuergerät eignen sich Zustandsautomaten. Damit kann der häufig komplexe Ablauf

übersichtlich festgelegt werden, wobei auch die geforderten Sicherheits- und Zuverlässigkeitsanforderungen berücksichtigt werden können.

1.1 Löschen und Programmieren von Flash-Speichern

Mit den derzeit eingesetzten Flash-Technologien können nur ganze Flash-Bereiche gelöscht oder neu programmiert werden. Die kleinste, physikalisch zusammen gehörende, geschlossen lösch- oder programmierbare Speichereinheit des Flash-Speichers wird Segment genannt. Bei der Flash-Programmierung sind deshalb die Schritte Löschen und Programmieren von Flash-Segmenten zu unterscheiden.

Des weiteren muss beachtet werden, dass aus technischen Gründen nicht gleichzeitig aus einem Flash-Segment ein Programm ausgeführt werden kann, während ein anderes Flash-Segment des gleichen Bausteins neu programmiert wird. Die Programmteile zur Steuerung des Programmierablaufs für einen Flash-Baustein müssen deshalb – zumindest temporär während der eigentlichen Flash-Programmierung – in einen anderen Speicherbaustein – beispielsweise in einem anderen Flash-Baustein oder einen freien RAM-Bereich des Mikrocontrollers – ausgelagert werden.

Außer diesen Randbedingungen werden in den folgenden Abschnitten weitere mikrocontroller- oder speicherspezifische Detailunterschiede nicht betrachtet.

1.2 Flash-Programmierung über die Off-Board-Diagnoseschnittstelle

Wegen der begrenzten Übertragungsleistung der Off-Board-Diagnoseschnittstelle kommt es bei großen Flash-Speichern zu recht langen Flash-Programmierzeiten. Deshalb besteht in der Produktion und im Service häufig die Anforderung, die Flash-Programmierzeiten zu verkürzen, was z. B. durch die Verringerung der neu zu programmierenden Flash-Segmente möglich ist. Dies kann durch die Flash-Programmierung einzelner Software-Funktionen oder durch die getrennte Flash-Programmierung für den Programm- und Datenstand des Steuergeräts erreicht werden. Daher wird häufig der Programmstand bereits bei der Steuergeräteproduktion programmiert, während der Datenstand später fahrzeugspezifisch am Ende der Fahrzeugproduktion programmiert wird. Für die Software-Entwicklung bedeutet dies, dass verschiedene Software-Funktionen, sowie Programm- und Datenstand in verschiedenen Flash-Segmenten abgelegt werden müssen.

Alle Programmteile des Mikrocontrollers, die für die Kommunikation zwischen Mikrocontroller und Flash-Programmierwerkzeug über die Off-Board-Diagnoseschnittstelle während der Flash-

Programmierung erforderlich sind (Bild 1), müssen zusammen mit den Flash-Programmierroutinen, dem so genannten Flash-Loader, im ROM oder in einem anderen weiteren Flash-Segment abgelegt werden.

Siehe Bild 1: Software-Architektur für Steuergeräte

In den folgenden Abschnitten wird der im ROM abgelegte Basisumfang als **Start-Up-Block** und der im Flash abgelegte Basisumfang als **Boot-Block** bezeichnet.

In Bild 2 ist die Organisation des gesamten Programms in diese vier Teile dargestellt. Start-Up- und Boot-Block zusammen stellen die für die Flash-Programmierung über die Off-Board-Diagnoseschnittstelle notwendige Software-Funktionalität des Mikrocontrollers zur Verfügung. Die Aufteilung in Start-Up- und Boot-Block ist aus verschiedenen Gründen sinnvoll. So kann der Boot-Block selbst, falls er im Flash abgelegt wird, neu programmiert werden. Darauf wird in Abschnitt 1.5 eingegangen. Außerdem kann im Boot-Block der aktuelle Status der Flash-Programmierung unverlierbar abgespeichert werden, so dass beispielsweise nach einem Abbruch ein Wiederaufsetzen möglich ist. Die unveränderbare Basisfunktionalität des Start-Up-Blocks und eine Kennung für die Hardware-Variante des Steuergeräts kann dagegen im kostengünstigeren und nicht neu programmierbaren ROM abgelegt werden.
Das Fahrprogramm, als Teil des Programm- und Datenstands, hingegen wird in einem anderen Speichersegment abgelegt. In den folgenden Abschnitten wird zwischen den folgenden Programmteilen unterschieden:

- Start-Up-Block
- Boot-Block
- Programmstand
- Datenstand.

Siehe Bild 2: Speicherzuteilung für Start Up-Block, Boot-Block, Programm- und Datenstand

1.3 Sicherheitsanforderungen

Der Übergang des Mikrocontrollers in den Betriebszustand „Software-Update" wird vom Flash-Programmierwerkzeug angestoßen.

Neben eventuell notwendigen Plausibilitätsprüfungen – etwa bei Motorsteuergeräten die Prüfung auf Motorstillstand –, die vor Beenden des Fahrprogramms und dem Übergang in den Betriebszustand „Software-Update" durchgeführt werden müssen, sind beim Einsatz in der Produktion und im Service weitere Sicherheitsmaßnahmen erforderlich. Aus Haftungsgründen muss eine nicht autorisierte Flash-Programmierung oder eine Flash-Programmierung mit manipuliertem Programm- oder Datenstand möglichst verhindert, auf jeden Fall aber erkannt und nachgewiesen werden können.

Daher wird der Flash-Programmierzugriff in der Regel über zwei unterschiedliche Verschlüsselungsverfahren abgesichert: die Authentisierung und die Signaturprüfung.

Der Ablauf der Kommunikation zwischen Flash-Programmierwerkzeug und Mikrocontroller ist in Bild 3 dargestellt:

- **Authentisierung**

Nach der Plausibilitätsprüfung wird die Prüfung der eigentlichen Zugriffsberechtigung durchgeführt. Dieser Schritt wird Authentisierung genannt. Dabei wird anhand eines digitalen Schlüssels überprüft, ob der Anwender des Flash-Programmierwerkzeugs überhaupt zum Software-Update berechtigt ist.

- **Signaturprüfung für den neu zu programmierenden Programm- oder Datenstand**

In einem weiteren Prüfungsschritt wird die Datenkonsistenz des neu zu programmierenden Programm- oder Datenstands überprüft. Dieser Schritt wird auch Signaturprüfung genannt. Hierbei wird vom Flash-Programmierwerkzeug anhand eines weiteren digitalen Schlüssels überprüft, ob der neu zu programmierende Programm- oder Datenstand zur Steuergeräte-Hardware passt und ob der neu zu programmierende Programm- oder Datenstand seit der Auslieferung durch den Fahrzeughersteller an das Werk oder die Servicewerkstatt unzulässig manipuliert wurde.

• **Löschen und Programmieren von Flash-Segmenten**
Erst nach erfolgreichem Abschluss beider Prüfungen wird das eigentliche Löschen und Programmieren der entsprechenden Flash-Segmente durch den Boot-Block freigegeben.

• **Signaturprüfung für den neu programmierten Programm- und Datenstand**
Nach der Flash-Programmierung wird die Signatur vom Mikrocontroller auf Basis des tatsächlich ins Flash programmierten Programm- und Datenstands berechnet, um Fehler während der Programmierung erkennen zu können. Nach erfolgreicher Signaturprüfung wird diese berechnete Signatur selbst im Flash abgelegt. Dazu werden besondere Speicherstrukturen, die sogenannte Programmstands- und Datenstands-Logistik als Teil des Programm- und des Datenstands im Flash abgelegt (Bild 4). Nur nach erfolgreicher Signaturprüfung gibt der Boot-Block die Aktivierung des neuen Fahrprogramms frei.

Siehe Bild 3: Sicherheitsmaßnahmen zum Schutz der Flash-Programmierung vor Missbrauch

Und Siehe Bild 4: Hardware-, Programmstands- und Datenstandslogistik für Berechnung, Ablage und Prüfung der Signatur

Auf die möglichen Verschlüsselungsverfahren soll dieser Stelle nicht weiter eingegangen werden.

1.4 Verfügbarkeitsanforderungen

Da die Flash-Programmierung über die Off-Board-Diagnoseschnittstelle trotz der angesprochenen Optimierungsmaßnahmen eine verhältnismäßig lange Zeitspanne in Anspruch nehmen kann, ist mit Abbrüchen des Programmierablaufs durch Störungen jederzeit zu rechnen. Derartige Störungen sind etwa der Ausfall der Spannungsversorgung des Fahrzeugs oder des Flash-Programmierwerkzeugs, unzulässige Reaktionen anderer Steuergeräte im Netzwerk, Unterbrechungen der Kommunikationsverbindung zwischen Steuergerät und Flash-Programmierwerkzeug oder Bedienfehler. Auch fehlgeschlagene Authentisierungen und Signaturprüfungen führen zum Abbruch der Flash-Programmierung.
Für den Entwurf des Ablaufs der Flash-Programmierung steht deshalb die Verfügbarkeit dieser Funktion unter allen denkbaren Umständen an erster Stelle. Diese Anforderung kann

beispielsweise dadurch erfüllt werden, dass nach einem Abbruch in allen Situationen jederzeit ein Neustart des Programmierablaufs möglich ist. Eine dafür geeignete Vorgehensweise wird im folgenden Beispiel durch einen Zustandsautomaten dargestellt.

Beispiel: Ablauf der Flash-Programmierung für Programm- und Datenstand

Bild 5 zeigt einen möglichen Ablauf für die Flash-Programmierung des in Bild 4 dargestellten Programm- und Datenstands.

Nach erfolgreichem Abschluss der Flash-Programmierung wird über einen Reset vom Flash-Programmierwerkzeug der Übergang des Mikrocontrollers in den normalen Betriebszustand angestoßen.

Siehe Bild 5: Zustände und Übergänge des Boot-Blocks bei der Flash-Programmierung von Programm- und Datenstand

Die notwendige Auslagerung des Boot-Blocks in einen anderen Speicherbaustein während der eigentlichen Flash-Programmierung wurde bisher vernachlässigt. Darauf und auf die Flash-Programmierung des Boot-Blocks selbst wird im folgenden Abschnitt eingegangen.

1.5 Auslagerung und Flash-Programmierung des Boot-Blocks

Abschließend soll ein Verfahren zur Flash-Programmierung des Boot-Blocks ausführlich dargestellt werden, wobei die bereits angesprochenen Randbedingungen der eingesetzten Flash-Technologie und die Verfügbarkeitsanforderungen berücksichtigt werden.

Zunächst muss der aktive Boot-Block während der Flash-Programmierung in einen anderen Speicherbaustein des Mikrocontrollers ausgelagert werden, d. h. der Boot-Block muss relocatierbar sein. Dies kann beispielsweise durch Kopieren des Boot-Blocks in einen während der Flash-Programmierung freien RAM-Bereich erfolgen. Anschließend wird dann der Boot-Block aus dem RAM ausgeführt.

Auch nach fehlgeschlagener Flash-Programmierung des Boot-Blocks muss ein Neustart des Programmierablaufs möglich sein. Zur Erhaltung der Verfügbarkeit ist nach einem Abbruch ein

fehlerfreier Boot-Block ausreichend. Diese Anforderung kann durch „Retten" und „Wiederherstellen" des Boot-Blocks erfolgen.

Beispiel: Ablauf der Flash-Programmierung für den Boot-Block

Diese Anforderungen können mit einem Ablauf, wie in Bild 6 dargestellt, erfüllt werden. Dabei wird zwischen altem und neuem Boot-Block unterschieden.

Siehe Bild 6: Schritte bei der Flash-Programmierung des Boot-Blocks

Das Verfahren unterscheidet drei wesentliche Schritte:

- Schritt 1: Kopieren des alten Boot-Blocks in einen freien RAM-Bereich
- Schritt 2.1: Aktivierung des alten Boot-Blocks im RAM und Deaktivierung des alten Boot-Blocks im Flash
- Schritt 2.2: Zwischenablage des neuen Boot-Blocks in Flash-Segment C

Dieser Schritt umfasst die Zustände Löschen des Flash-Segments C, Programmieren des neuen Boot-Blocks in Flash-Segment C und Signaturprüfung für den neuen Boot-Block in Flash-Segment C.

Nach einem Abbruch während dieser Operationen kann mit dem gültigen, alten Boot-Block in Flash-Segment A die Flash-Programmierung erneut gestartet werden.

- Schritt 3: Programmieren des neuen Boot-Blocks durch Kopieren von Flash-Segment C nach Flash-Segment A

Dieser Schritt umfasst die Zustände Löschen des Flash-Segments A, Programmieren des neuen Boot-Blocks in Flash-Segment A durch Kopieren des Flash-Segments C nach A und Signaturprüfung für den neuen Boot-Block in Flash-Segment A.

Nach einem Abbruch während dieser Operationen kann mit dem gültigen, neuen Boot-Block in Flash-Segment C die Flash-Programmierung erneut gestartet werden.

Der jeweils gültige Boot-Block im Flash muss markiert werden. Diese Gültigkeitsmarkierung selbst muss unverlierbar im Flash abgelegt werden, so dass mit dieser Information ein Wiederaufsetzen möglich ist.

Anschließend erfolgt die Aktivierung des neuen Boot-Blocks in Flash-Segment A und die Deaktivierung des Boot-Blocks im RAM. Danach muss die Flash-Programmierung für den Datenstand, wie in Bild 5 beschrieben, erfolgen.

Vorgehensweise und Kerngedanken der Erfindung

Schritt 1: Festlegung der Anforderungen an die Flash-Programmierung

1.1 Spezifikation der Software-Betriebszustände des Steuergeräts, z. B. der Zustände „Anfangszustand", „Normalbetrieb" und „Software-Update", sowie der möglichen Übergänge zwischen diesen Betriebszuständen und der jeweiligen Übergangsbedingungen.

1.2 Spezifikation der für die Flash-Programmierung relevanten Speicherblöcke der Software des Steuergeräts, insbesondere Zuordnung der neu zu programmierenden Software-Komponenten zu den Speicherblöcken Start-Up-Block, Boot-Block, Programm- und Datenstand, ggf. weitere Unterteilung in Subblöcke.

1.3 Zuordnung der Speicherblöcke zu den Speichern des Mikrocontrollers des Steuergeräts, also z. B. zu den ROM- und Flash-Bausteinen.

1.4 Spezifikation der Sicherheits-, Zuverlässigkeits- und Verfügbarkeitsanforderungen, z. B. der notwendigen Verschlüsselungs- und Prüfungsverfahren für die Authentisierung und Signaturprüfung vor und nach dem Flash-Programmiervorgang, und einer Strategie zum Weiterbetrieb des Steuergeräts bzw. Neustart /Wiederaufsetzen des Flash-Programmiervorgangs nach Unterbrechungen/Abbruch eines Flash-Programmiervorgangs.

Schritt 2: Spezifikation des Ablaufs der Flash-Programmierung durch Zustandsautomaten

2.1 Spezifikation der erforderlichen Subzustände des Betriebszustandes „Software-Update", insbesondere der Subzustände „Abbruch/Fehlermeldung" und „Abschluss/Erfolgsmeldung", der Übergänge zwischen diesen Subzuständen und der Übergangsbedingungen, sowie eines Verfahrens zum dauerhaften, unverlierbaren Abspeichern des zuletzt gültigen oder fehlerfrei durchlaufenen Subzustandes.

2.2 ggf. Spezifikation von Subzuständen für Authentisierung und Signaturprüfung

2.3 ggf. Spezifikation von Subzuständen für die Auslagerung und Flash-Programmierung des Boot-

Blocks

2.4 Spezifikation von Subzuständen für das Löschen und Programmieren von Flash-Segmenten

2.5 Spezifikation der Übergänge zwischen den Subzuständen und der Übergangsbedingungen

Schritt 3: Überprüfung der Verfügbarkeits-, Sicherheits- und Zuverlässigkeitsanforderungen

für jeden Zustand und jeden Übergang des Zustandsautomaten bei erfolgreichem und erfolglosem/fehlerhaften Durchlaufen des Zustandes.

ggf. Wiederholung der Schritte 2.1 bis 2.5 und Korrektur des spezifizierten Ablaufs für die Flash-Programmierung.

20.06.03 Sy

ROBERT BOSCH GMBH, 70442 Stuttgart

Spezifikation des Ablaufs des Software-Updates für elektronische Steuergeräte durch Flash-Programmierung über serielle Schnittstellen

Zusammenfassung

Für die Spezifikation und den Test des Zusammenspiels zwischen Flash-Programmierwerkzeug und Steuergerät eignen sich Zustandsautomaten. Damit kann der häufig komplexe Ablauf übersichtlich festgelegt werden, wobei auch die geforderten Sicherheits- und Zuverlässigkeitsanforderungen berücksichtigt werden können.

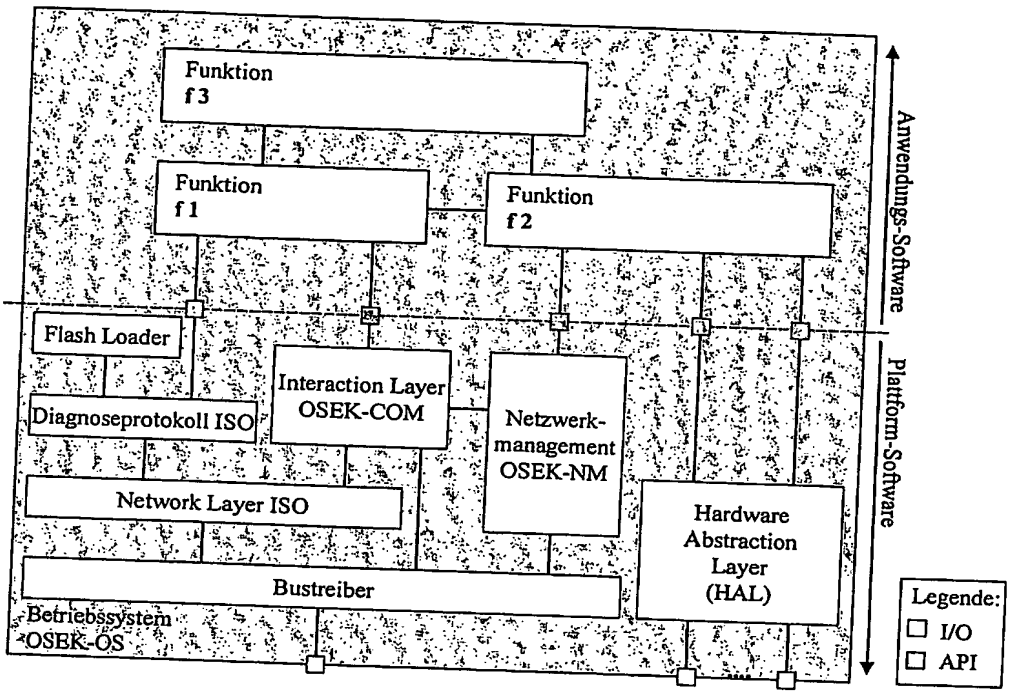

Bild 1: Software-Architektur für Steuergeräte

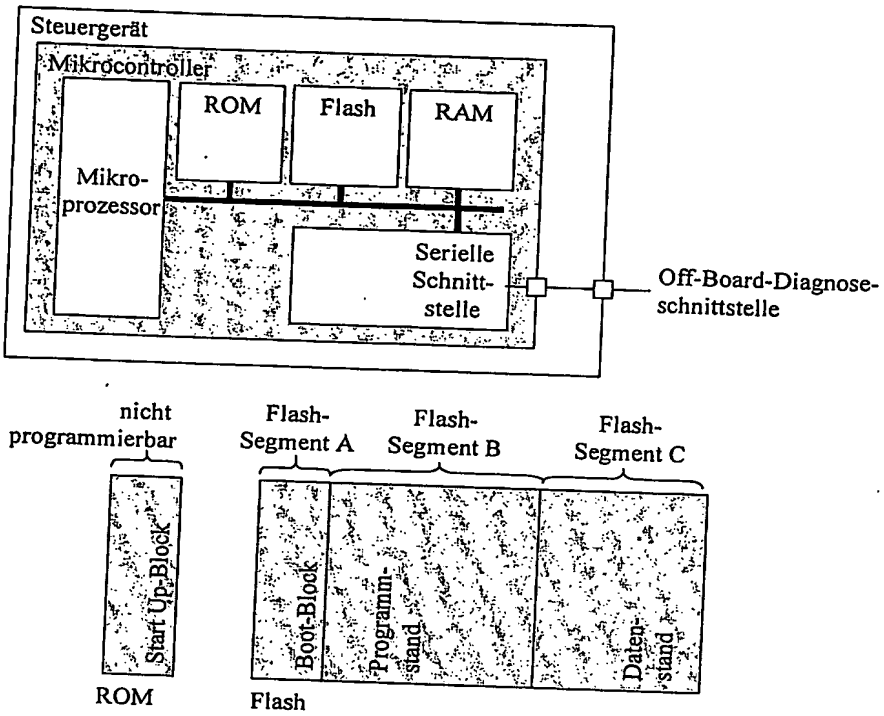

Bild 2: Speicherzuteilung für Start Up-Block, Boot-Block, Programm- und Datenstand

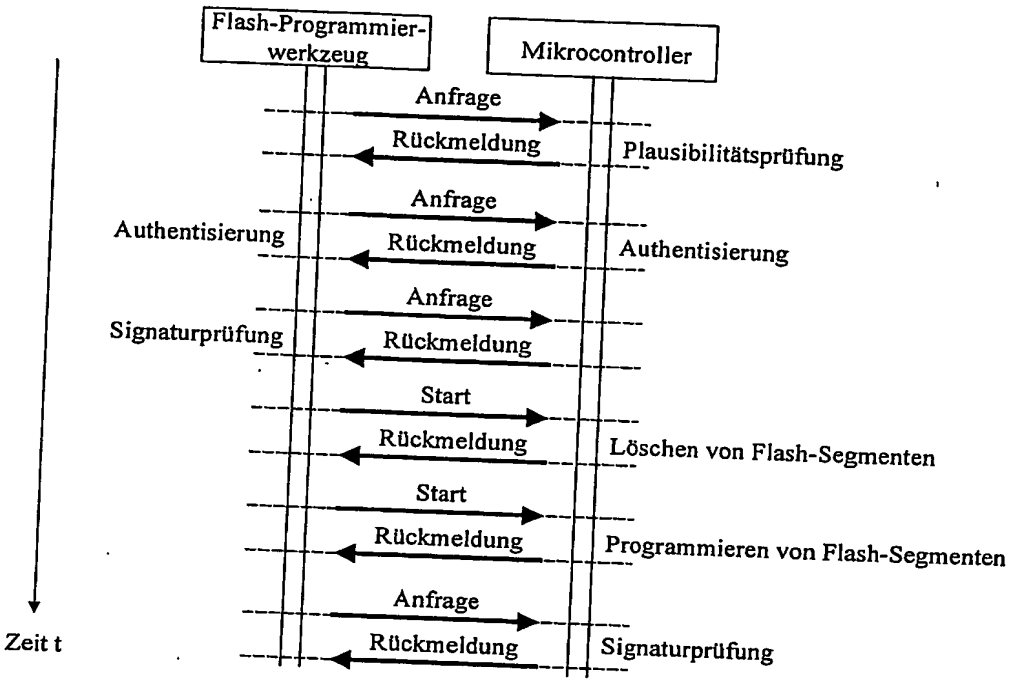

Bild 3: Sicherheitsmaßnahmen zum Schutz der Flash-Programmierung vor Missbrauch

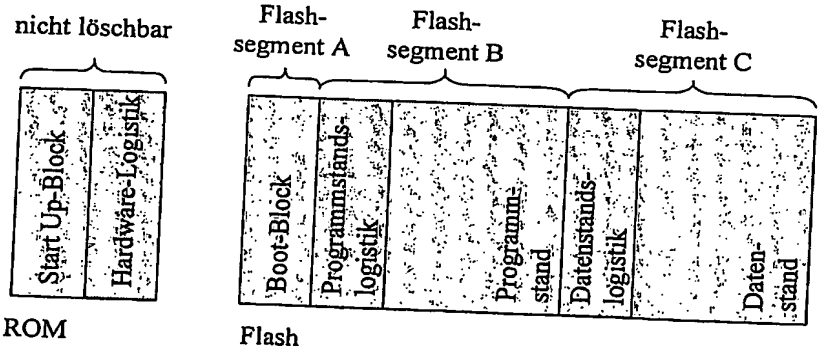

Bild 4: Hardware-, Programmstands- und Datenstandslogistik für Berechnung, Ablage und Prüfung der Signatur

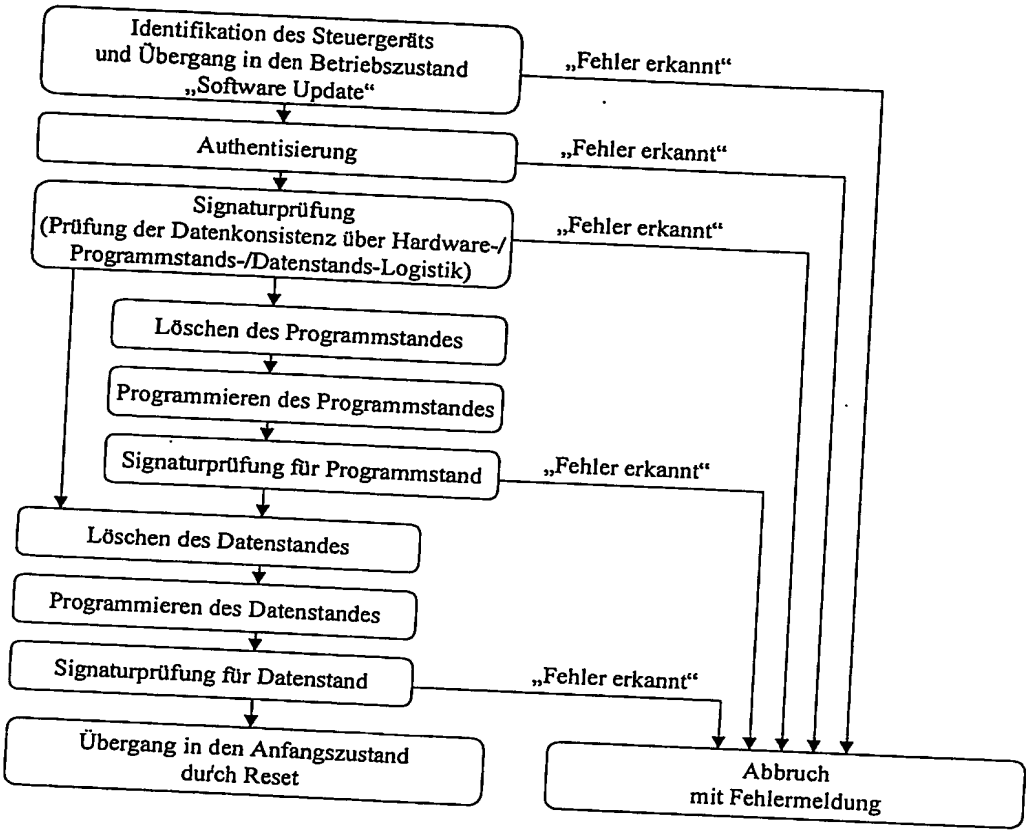

Bild 5: Zustände und Übergänge des Boot-Blocks bei der Flash-Programmierung von Programm- und Datenstand

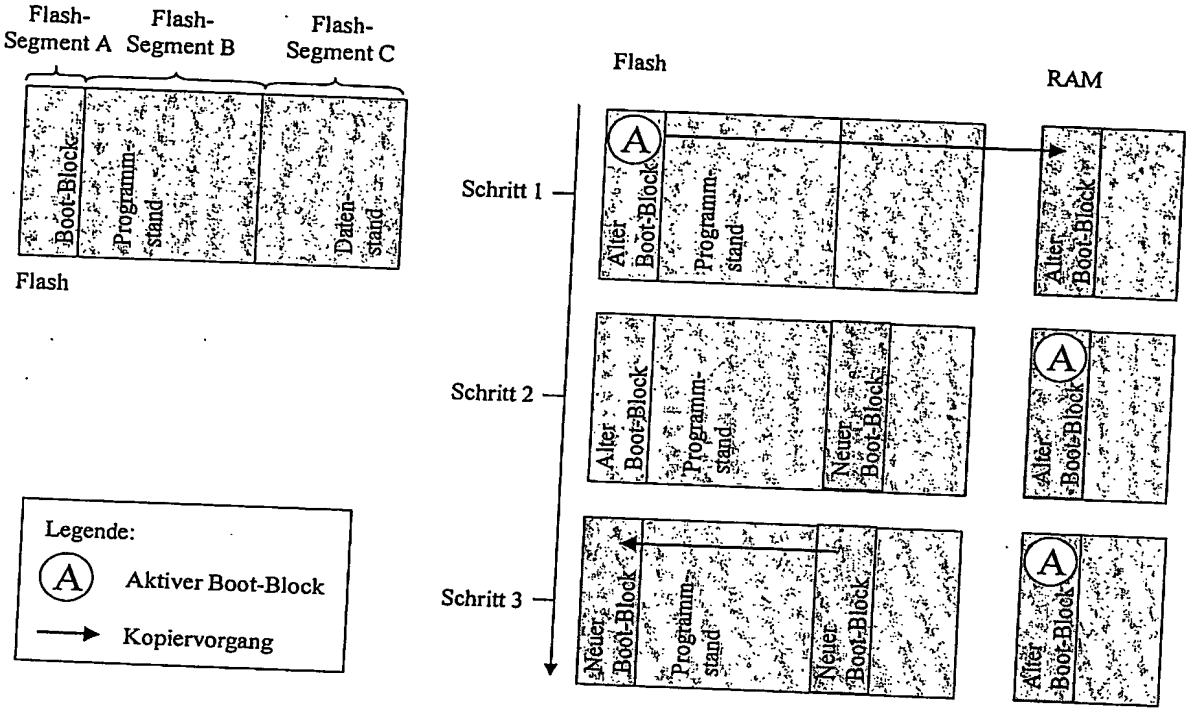

Bild 6: Schritte bei der Flash-Programmierung des Boot-Blocks